

Name:

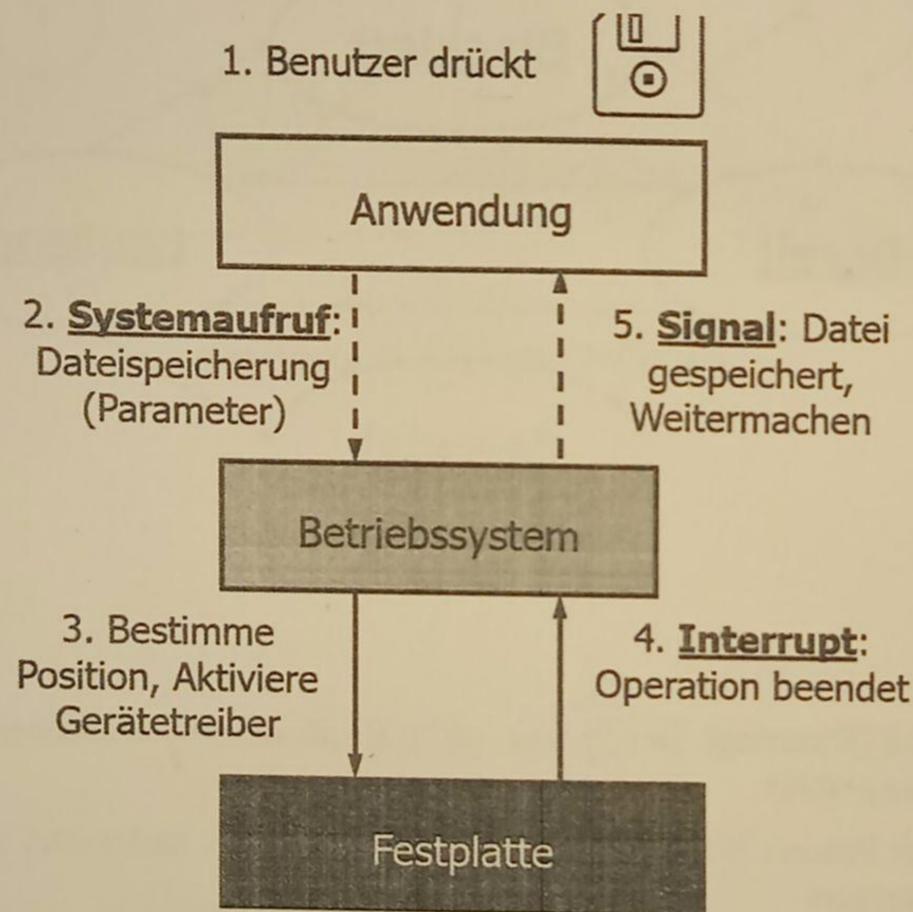
Korrektorexemplar

Matrikelnummer:

Aufgabe 1: Prozesse

(12 Punkte)

(a) (4 Punkte) Welche Kommunikationsschritte eines Betriebssystems sind nötig, um eine Datei auf der Festplatte zu speichern? Beschriften Sie die vier Pfeile in der Abbildung und beschreiben Sie unterhalb der Abbildung die Schritte **kurz und stichpunktartig**.



[0,5 Punkte pro korrekte Beschriftung; 0,5 Punkte pro korrekter Beschreibung]

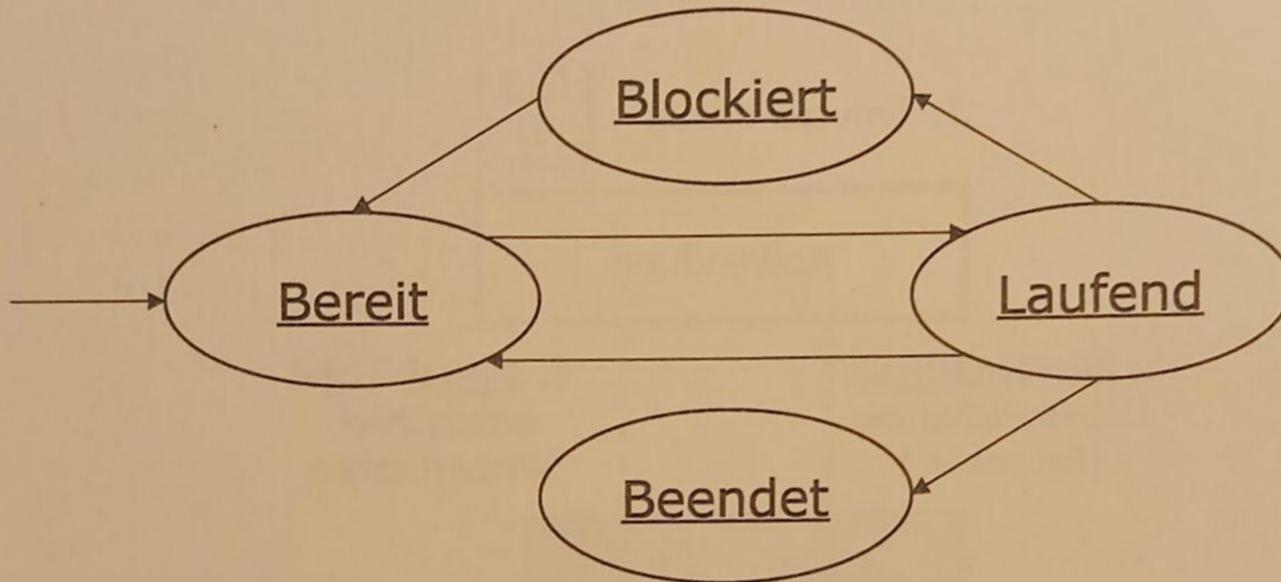
Punkte	
--------	--

Korrektorexemplar

Matrikelnummer:

Name:

(b) (2 Punkte) Nennen Sie mit Hilfe der folgenden Skizze vier mögliche Prozesszustände und beschreiben Sie diese mit je einem Satz. (Hinweis: Die Pfeile zwischen den Zuständen dienen nur als Orientierungshilfe, Sie müssen diese nicht beschriften.)



Rechnend, Laufend (Running) Der Prozess ist im Besitz des physikalischen Prozessors und wird aktuell ausgeführt

Bereit (Ready) Der Prozess hat alle notwendigen Betriebsmittel und wartet auf die Zuteilung des/eines Prozessors

Blockiert, Wartend (Waiting) Der Prozess wartet auf die Erfüllung einer Bedingung, z.B. Beendigung einer E/A-Operation und bewirbt sich derzeit nicht um den Prozessor

Beendet (Terminated) Der Prozess hat alle Berechnung beendet und die zugeteilten Betriebsmittel freigegeben

[0,5 Punkte pro korrekte Beschriftung, 0,5 Punkte pro korrekte Beschreibung]

Punkte	
--------	--

Korrektorexemplar

Name: _____

Matrikelnummer: _____

(c) (6 Punkte) Gegeben ist der folgende Abhängigkeitsgraph. Setzen Sie diesen mit Hilfe der aus der Übung bekannten Befehle **fork/join** und **parbegin/parend** in Pseudocode um.

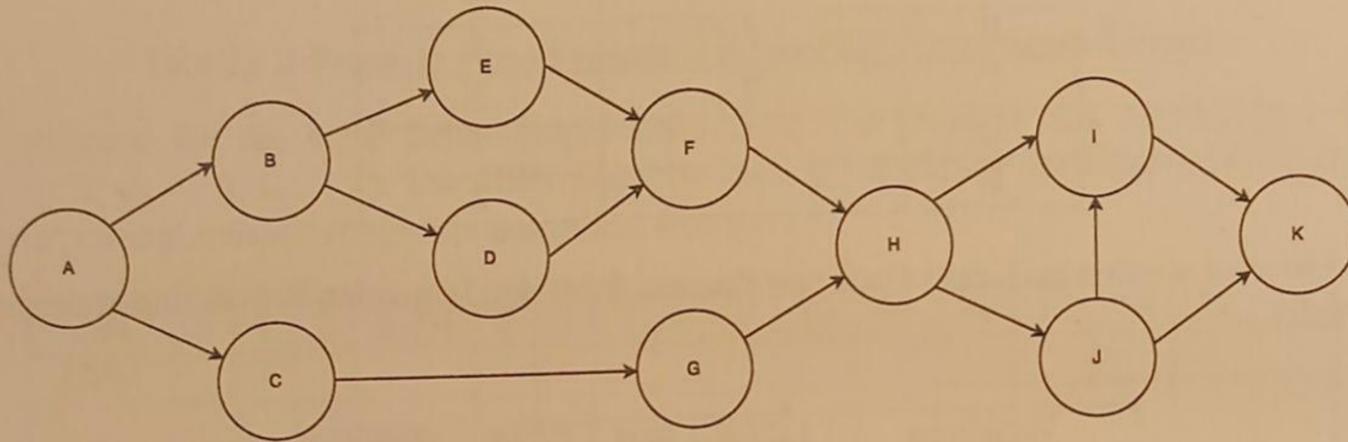


Abbildung 1: Abhängigkeitsgraph

```

A
fork(top)
C
G
join(top)
H
J
I
K

top:
B
fork(E)
D
join(E)
F

A
parbegin
begin
C
G
end
begin
B
parbegin
E
D
parend
F
end
parend
H
J
I
K
  
```

[1p pro richtigem fork, join, parbegin, parend, begin, end]

Punkte	
--------	--

Name:

(12 Punkte)

Aufgabe 2: Scheduling

(a) (2 Punkte) Die folgende Tabelle beschreibt:

Prozesse	Laufzeit (Dauer)	Ankunftszeit
A	6	0
B	4	5
C	8	3
D	3	7
E	5	2

Welche Prozesse werden nach dem Ende von Prozess A bei den folgenden Schedulingstrategien ausgewählt?

Verfahren	LCFS	SJN	FCFS	HRRN
Nächster Prozess				

Verfahren	LCFS	SJN	FCFS	HRRN
Nächster Prozess	B	B	E	E

[0.5 P. je Verfahren]

Punkte	
--------	--

Korrektorexemplar

Name: _____

Matrikelnummer: _____

(b) (4 Punkte)

Prozess	A	B	C	D
Ankunftszeitpunkt	0	2	4	6
Dauer	5	4	2	3

Tabelle 1: Prozesse eines Systems mit einer CPU und einem Thread

Simulieren Sie das MLF Scheduling-Verfahren für die Prozesse aus Abbildung 1 (RR mit $\tau_i = 2^i$ ($i = 0, 1, \dots$)). Die Hilfszeilen für LVL 0 - 3 dienen der Übersichtlichkeit für die Bearbeitung, müssen aber nicht ausgefüllt werden.

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CPU																	
LVL 0																	
LVL 1																	
LVL 2																	
LVL 3																	

Tabelle 2: Schedule nach Multilevel Feedback-Verfahren

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CPU																	
LVL 0																	
LVL 1																	
LVL 2																	
LVL 3																	

Tabelle 3: Ersatztabelle

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CPU	A	A	A	B	C	B	B	D	C	D	D	A	A	B			

Punkte	
--------	--

Name:

- (c) (6 Punkte) Konstruieren Sie basierend auf den Shortest Remaining Time Next- (SRTN) und Last Come First Served Preemptive Resume- (LCFS-PR) Verfahren für die Prozesse aus Tabelle 4 jeweils einen gültigen Schedule in den Tabellen ?? und ?. Tragen Sie anschließend jeweils Warte- und Antwortzeiten der beiden Verfahren in die Tabellen ?? und ?? ein. Hinweis: Bei Ankunft werden die Prozesse in eine FIFO queue eingefügt.

Prozesse	Laufzeit (Dauer)	Ankunftszeit
A	4	1
B	3	9
C	7	8
D	7	5
E	5	7

Tabelle 4: Aperiodische Prozesse

Zeit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CPU															
Zeit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
CPU															

	A	B	C	D	E
Wartezeit	0	3	12	0	8
Antwortzeit	4	6	19	7	13

Punkte	
--------	--

Korrektorexemplar

Name: _____

Matrikelnummer: _____

Prozesse	Laufzeit (Dauer)	Ankunftszeit
A	4	1
B	3	9
C	7	8
D	7	5
E	5	7

Tabelle 4: Aperiodische Prozesse (Kopie von Seite 7)

Zeit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CPU	A	A	A	A	D	D	E	C	B	B	B	D	D	D	D
Zeit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
CPU	D	E	E	E	E	C	C	C	C	C	C				
		A	B	C	D	E									
Wartezeit		0	0	12	5	9									
Antwortzeit		4	3	19	12	14									

Bewertung: Linear mit korrektem Anteil

SRTN: 2 P. Verfahren, 1 P. Warte- und Antwortzeiten

LCFS-PR: 2 P. Verfahren, 1 P. Warte- und Antwortzeiten

Punkte	
--------	--

Name:

Aufgabe 3: Koordination**(12 Punkte)**

- (a) (4 Punkte) Die Prozesse P1 und P2 werden nebenläufig ausgeführt. Geben Sie an, welche Werte die Variable x nach Beendigung der Prozesse annehmen kann. Es handelt sich **nicht** um atomare Operationen.

1 Ausgangszustand:

2 $x=2;$

3

4 P1:

5 $x = x * 3;$

6

7 P2:

8 $x++;$

$x = 7$

$x = 9$

$x = 6$

$x = 3$

[1p pro Ergebnis]

Punkte	
--------	--

Name:

Korrektorexemplar

Matrikelnummer:

(b) (8 Punkte)

In einem Prüfungsamt einer Universität gibt es fünf Schalter. Aus diesem Grund dürfen sich im Prüfungsamt maximal fünf Studierende gleichzeitig aufhalten.

Vervollständigen Sie unter Verwendung der *in der Vorlesung* vorgestellten Funktionen `signal`, `wait`, `lock` und `unlock` den Pseudocode des Prozesses `student`, um sicherzustellen, dass die maximale Anzahl von Studierenden im Prüfungsamt nicht überschritten wird.

```
// definieren Sie hier ggf. eigene globale Variablen
// (gelten fuer alle Prozesse vom Typ Student)
```

```
int counter;
```

```
    Mutex L;
    Signal C;
```

```
// (Funktion wird einmalig ausgefuehrt)
```

```
init(){
    counter = 0;
}
```

```
// Prozess student
```

```
// (wird fuer jede*n Studierende*n ausgefuehrt)
```

```
proc_student(){

    lock( L );
    while ( counter >= 5 ){
        unlock( L );
        wait( C );
        lock( L );
    }
    counter++;
    unlock( L );
```

```
//Studie betritt Pruefungsamt
```

```
enterPA();
```

```
//Studie verlaesst Pruefungsamt
```

```
    lock( L );
    counter--;
    unlock( L );
    signal( C );
```

```
}
```

[Linear mit korrektem Anteil]

Punkte	
--------	--

Name:

(4 Punkte)

Aufgabe 4: Betriebsmittelverwaltung

- (a) (4 Punkte) Gegeben sei ein System mit drei Prozessen und vier Betriebsmitteltypen. Der aktuelle Anforderungs- und Belegungszustand sei durch den Betriebsmittelgraphen in Abbildung 2 gegeben.

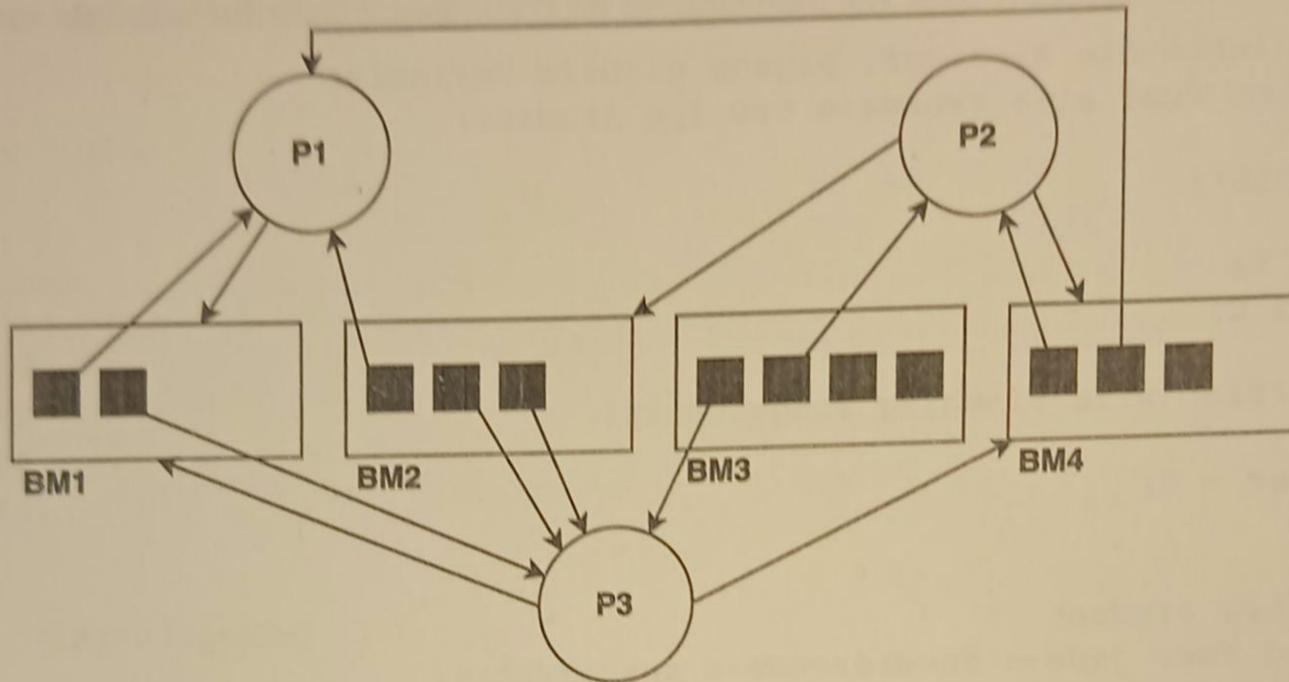


Abbildung 2: Betriebsmittelgraph (Resource-Allocation-Graph)

Geben Sie die dazugehörigen Matrizen für die belegten Betriebsmittel **B**, die Rest- und die Gesamtanforderung der Prozesse **R** bzw. **G**, den Vektor der freien Betriebsmittel **f** und den Vektor der vorhandenen Betriebsmittel **v** an.

B	BM1	BM2	BM3	BM4
P1	1	1	0	1
P2	0	0	1	1
P3	1	2	1	0

R	BM1	BM2	BM3	BM4
P1	1	0	0	0
P2	0	1	0	1
P3	1	0	0	1

G	BM1	BM2	BM3	BM4
P1	2	1	0	1
P2	0	1	1	2
P3	2	2	1	1

f	BM1	BM2	BM3	BM4
	0	0	2	1

v	BM1	BM2	BM3	BM4
	2	3	4	3

[1 Punkt pro richtiger Matrize. 0,5 Punkte pro Vektor]

Punkte	
--------	--

Name:

Korrektorexemplar

Matrikelnummer:

Aufgabe 5: Speicherverwaltung

(4 Punkte)

- (a) (4 Punkte) Gegeben ist der folgende Ausschnitt des Hauptspeichers inklusive Indizes. Der Byte-adressierte Speicher ist 2^8 Byte groß und seine Seitentabelle, die im gleichen Speicher liegt, enthält 2^4 Einträge. Die Tabellenbasisadresse ist 0x32.

Adresse	Wert
...	...
0x31	0x1E
0x32	0x0F
0x33	0x0D
0x34	0x0C
0x35	0x0E
0x36	0x0B
0x37	0x0A
0x38	0x08
0x39	0x09
0x3A	0x06
0x3B	0x01
0x3C	0x04
0x3D	0x08
0x3E	0x02
0x3F	0x07
0x40	0x03
0x41	0x05
...	...

Geben Sie zu den folgenden virtuellen Adressen die dazugehörigen physischen Adressen an. Geben Sie Ihre Antwort als Hexadezimal mit dem Prefix 0x und zwei folgenden Ziffern in Großbuchstaben an.

virtuelle Adresse	physische Adresse
0x0C	0xFA
0x70	0x94
0xC1	0x71
0xE1	0xF2

Punkte	
--------	--

Name:

Korrektorexemplar
Matrikelnummer:

Aufgabe 6: Dateisysteme

(6 Punkte)

- (a) (2 Punkte) Mit welcher Datenstruktur speichert ein Indexblock die Blockliste? Welcher Vorteil ergibt sich daraus im Vergleich zu einem FAT System? Antworten Sie in zwei Sätzen!

Indexblock: enthält Blockliste für diese Datei (als Array, nicht verkettet – daher auch schneller Zugriff an beliebige Positionen der Datei)

Zerstörung des Indexblocks -> Verlust nur dieser einen Datei

[2P]

- (b) (4 Punkte) Gegeben ist ein blockbasiertes Dateisystem. Der nachfolgenden Tabelle ist die Belegung der Festplattenpartition durch drei Dateien A, B, C zu entnehmen. Die Blöcke der Dateien sind auf der Partition verteilt und werden mit einem Index beginnend bei 1 durchnummeriert. So bedeutet der Speicherinhalt B2 beispielsweise den zweiten Block der Datei B.

Block	Speicherinhalt
0	C2
1	B3
2	C3
3	-
4	A5
5	A2
6	A6
7	A7
8	B4
9	C1
10	A4
11	C4
12	B1
13	-
14	A1
15	B2
16	A3

Tabelle 6: Speicherbelegung

Punkte	
--------	--

Name:

Korrektorexemplar

Matrikelnummer:

Geben Sie zu dieser Speicherbelegung die zugehörige FAT (File Allocation Table) wie in der Vorlesung besprochen an. Markieren Sie das Dateiende sowie einen nicht belegten Speicherinhalt (-) durch *NULL*.

Block	nächster Block
0	2
1	8
2	11
3	NULL
4	6
5	16
6	7
7	NULL
8	NULL
9	0
10	4
11	NULL
12	15
13	NULL
14	5
15	1
16	10

Punkte	
--------	--